

Date: October 6, 2003

To: Examiner John Q. Chavis, TC/A.U. 2124

Fax: (703) ⁷⁴⁶⁻⁷²⁴⁰~~308-3988~~

☐ Use this fax
number only

Company: U.S. Patent and Trademark Office

Phone: (703) 305-9665

☐ Notify recipient
before sending

From: George A. Willman

Phone: 650.320.4945

Return Fax: 650.493.6811

Original: ☐ To follow via mail ☐ To follow via courier ☐ To follow via email ☒ Original will not follow

Fax Contains: 3 pages (including this sheet). If incomplete, call 650.493.9300 x7255.

Message:

Please find attached an agenda for the teleconference Tuesday (October 7, 2003) regarding application number 09/347,473.

Ref: 17201.707

Return Original to: Linda Faye

Location: SR 1-2-i

650 Page Mill Road, Palo Alto, CA 94304-1050 • 650.493.9300 Tel • 650.493.6811 Fax • www.wsgr.com

This fax may contain confidential and privileged material for the sole use of the intended recipient. Any review or distribution by others is strictly prohibited.
If you are not the intended recipient please contact the sender and destroy all copies.

Entire Transmission Copyright © 2003 Wilson Sonsini Goodrich & Rosati. All Rights Reserved.

AGENDA FOR INTERVIEW

Appl. No. : 09/347,473 Confirmation No. 3911
Applicant : Philippe Harscoet
Filed : July 2, 1999
TC/A.U. : 2124
Examiner : Matthew P. Gubiotti

Docket No. : 17201.707
Customer No. : 21971

Rejection of Claim 1 under 35 USC 102 (e)

Discuss the application of Tock in rejection of claim 1. The Office Action indicates that Tock teaches a method substantially as claimed for loading Java™ classes as needed at run-time (in the Office Action mailed 7/30/03, p. 5).

It is believed that the characterization of Tock in the Office Action is not correct, and that in particular, Tock does not teach:

for data structures in a set of data structures, as unloaded data structures are needed during runtime,

receiving a data structure from a first memory, the data structure including one or more sets of instructions and one or more constants;

storing instructions from the data structure in a first portion of a second memory, the second memory comprising RAM;

storing constants from the data structure in a second portion of the second memory if only if the respective constant has not been stored in the second portion of the second memory,

modifying indexes in instructions that reference the constants to correspond to the respective locations of the constants in the second portion of the second memory, and

reading and executing at least some instructions from the data structure from the RAM.

In particular, Tock teaches an “offline class loader,” and thus teaches away from the approach claimed in claim 1 which involves activities performed for data structures in a set of data structures, as unloaded data structures are needed during runtime.

Rejection of Claim 16

Discuss possible alternate claim language (changes shown with underline):

16. (original): A computer system comprising:

a memory;
a virtual machine;
first logic that, after the virtual machine has been started, for classes in a set of
classes,
 receives a class from a class file, the class including one or more methods and
 one or more constants;
 stores instructions from the class in a first portion of the memory;
 stores constants from the class in a second portion of the memory if only if the
 respective constant has not been stored in the second portion of the memory, and
 modifies indexes within methods that reference the constants to correspond to the
respective locations of the constants in the second portion of the memory; and
 second logic that executes methods stored in the memory;
 wherein the memory, the first logic, and the second logic are coupled locally.

Rejection of Claims for use of “Java”

Discuss alternate language.